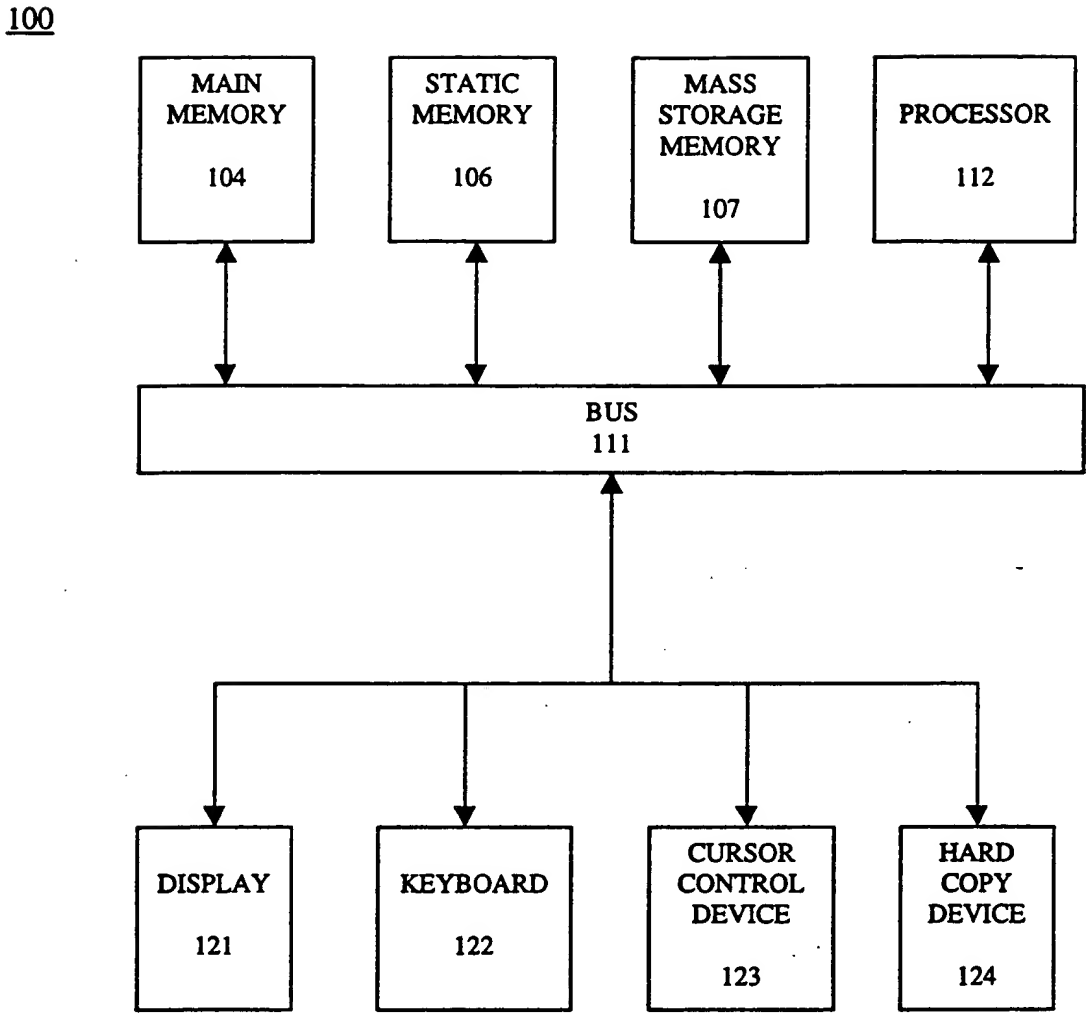

100
MAIN MEMORY 104
STATIC MEMORY 106
MASS STORAGE MEMORY 107
PROCESSOR 112
BUS 111
DISPLAY 121
KEYBOARD 122
CURSOR CONTROL DEVICE 123
HARD COPY DEVICE 124

FIG. 1

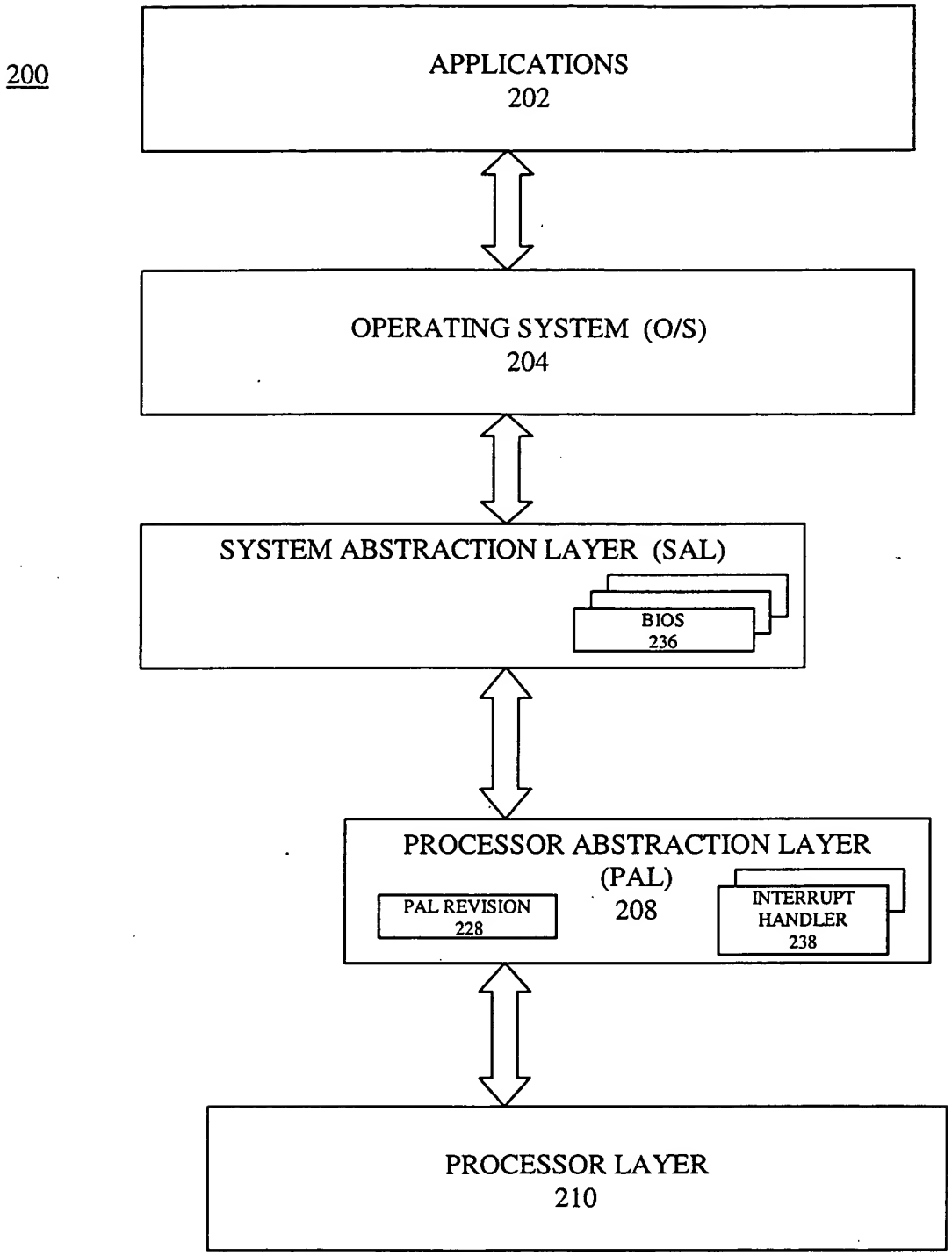

200
APPLICATIONS
202
OPERATING SYSTEM (O/S)
204
SYSTEM ABSTRACTION LAYER (SAL)
BIOS
236
PROCESSOR ABSTRACTION LAYER
(PAL)
208
PAL REVISION
228
INTERRUPT
HANDLER
238
PROCESSOR LAYER
210

FIG. 2

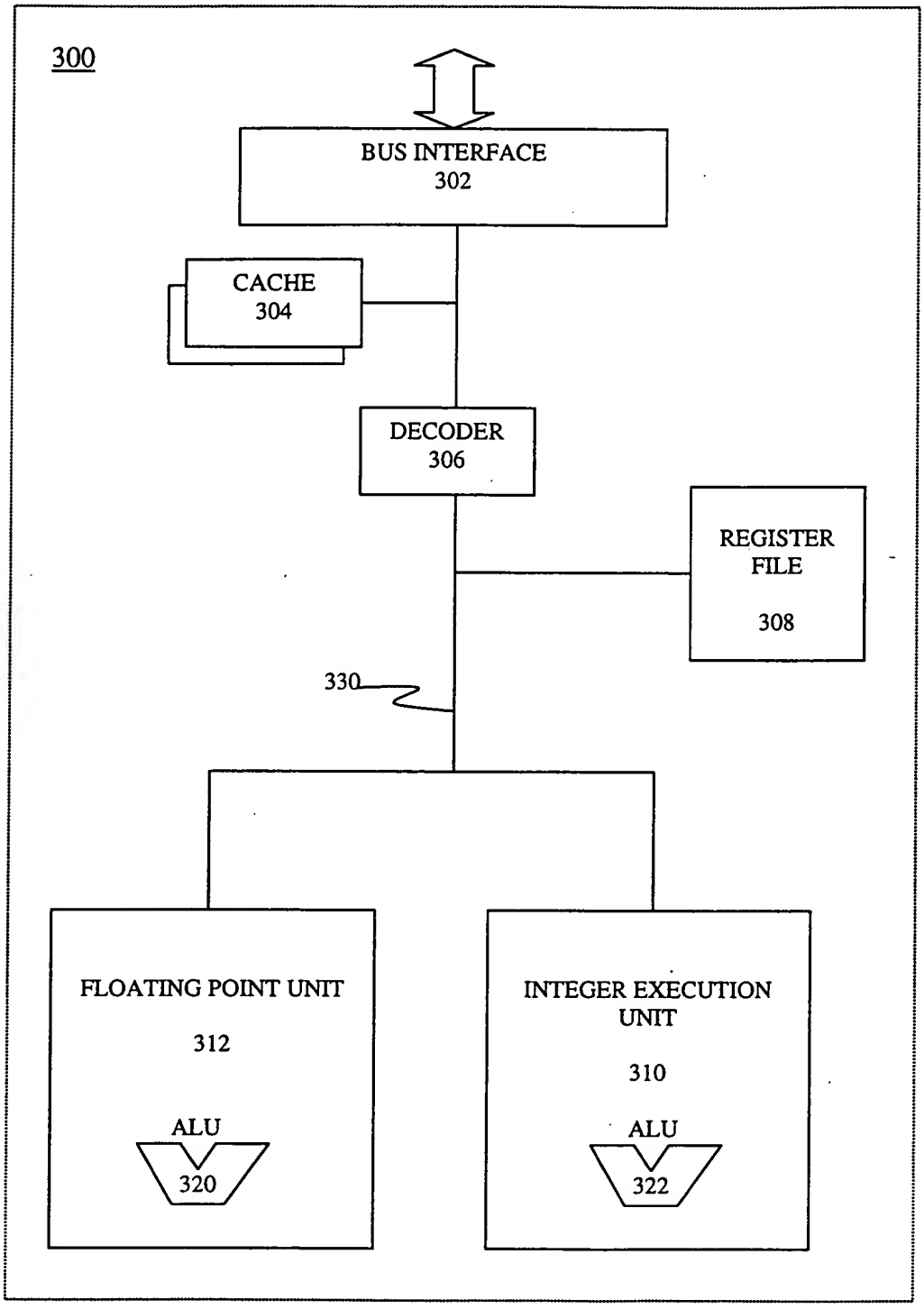

300
BUS INTERFACE
302
CACHE
304
DECODER
306
REGISTER
FILE
308
330
FLOATING POINT UNIT
312
ALU
320
INTEGER EXECUTION
UNIT
310
ALU
322

FIG. 3

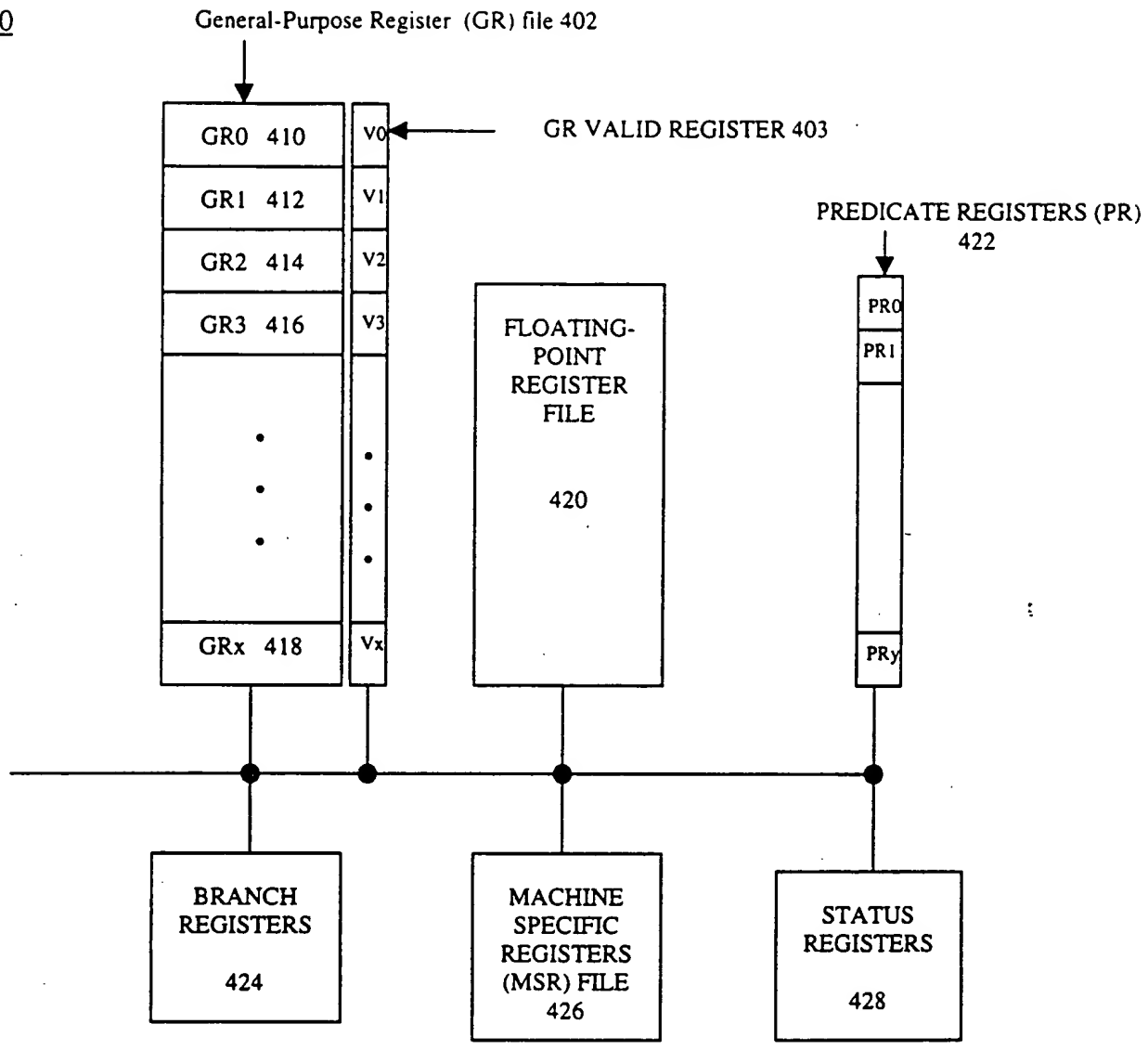

400
General-Purpose Register (GR) file 402
GR0 410
GR1 412
GR2 414
GR3 416
GRx 418
V0
V1
V2
V3
Vx
GR VALID REGISTER 403
FLOATING-POINT REGISTER FILE
420
PREDICATE REGISTERS (PR)
422
PR0
PR1
PRy
BRANCH REGISTERS
424
MACHINE SPECIFIC REGISTERS (MSR) FILE
426
STATUS REGISTERS
428

FIG. 4

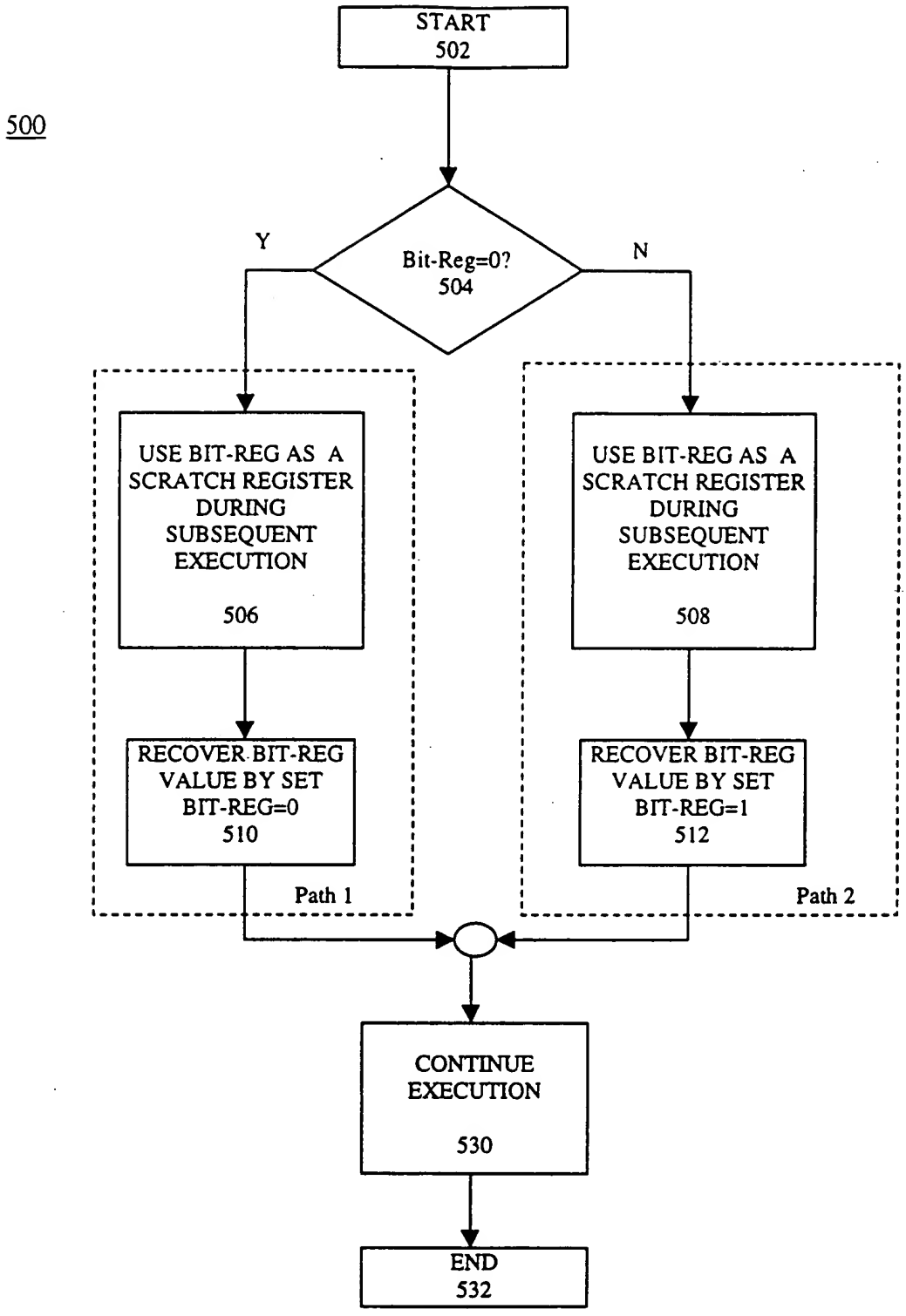

500
START
502
Bit-Reg=0?
504
Y
N
USE BIT-REG AS A SCRATCH REGISTER DURING SUBSEQUENT EXECUTION
506
USE BIT-REG AS A SCRATCH REGISTER DURING SUBSEQUENT EXECUTION
508
RECOVER BIT-REG VALUE BY SET BIT-REG=0
510
RECOVER BIT-REG VALUE BY SET BIT-REG=1
512
Path 1
Path 2
CONTINUE EXECUTION
530
END
532

FIG. 5

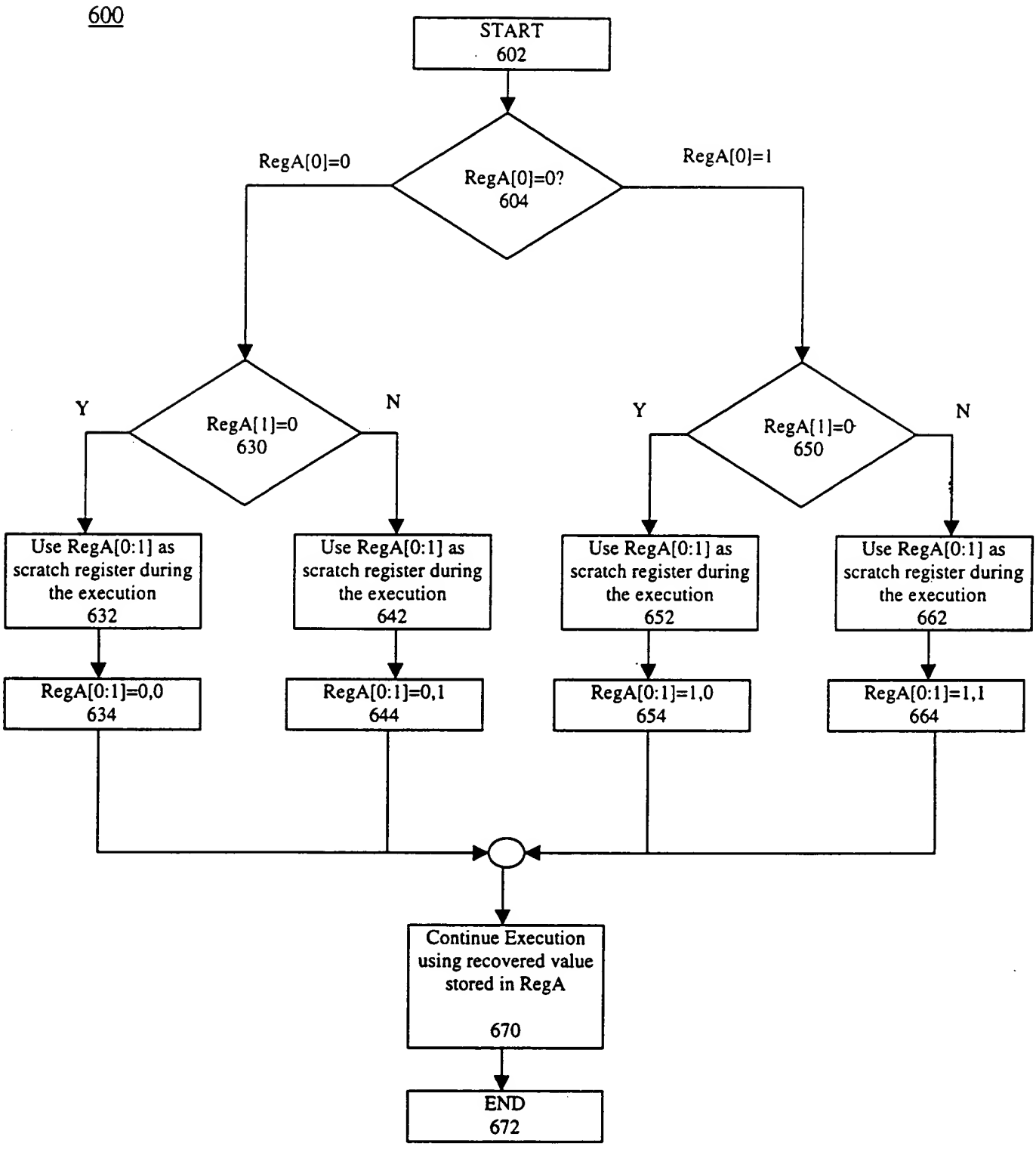

600
START
602
RegA[0]=0?
604
RegA[0]=0
RegA[0]=1
RegA[1]=0
630
Y
N
RegA[1]=0
650
Y
N
Use RegA[0:1] as scratch register during the execution
632
Use RegA[0:1] as scratch register during the execution
642
Use RegA[0:1] as scratch register during the execution
652
Use RegA[0:1] as scratch register during the execution
662
RegA[0:1]=0,0
634
RegA[0:1]=0,1
644
RegA[0:1]=1,0
654
RegA[0:1]=1,1
664
Continue Execution using recovered value stored in RegA
670
END
672

FIG. 6

700

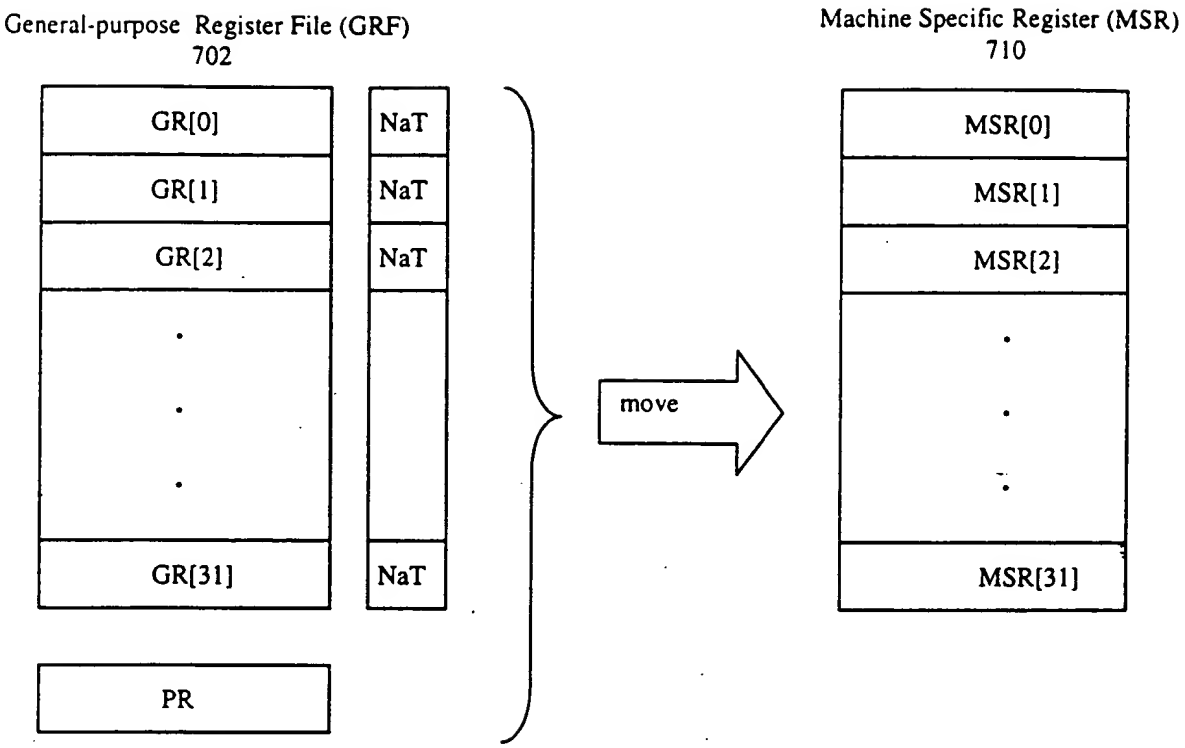

General-purpose Register File (GRF)
702
GR[0]
GR[1]
GR[2]
GR[31]
NaT
NaT
NaT
NaT
PR
move
Machine Specific Register (MSR)
710
MSR[0]
MSR[1]
MSR[2]
MSR[31]

FIG. 7

800

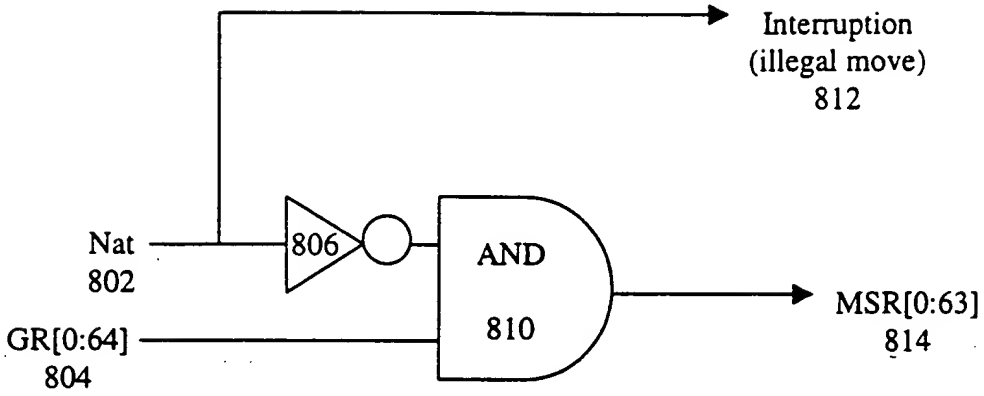

Interruption
(illegal move)
812
Nat
802
806
AND
810
MSR[0:63]
814
GR[0:64]
804

FIG. 8

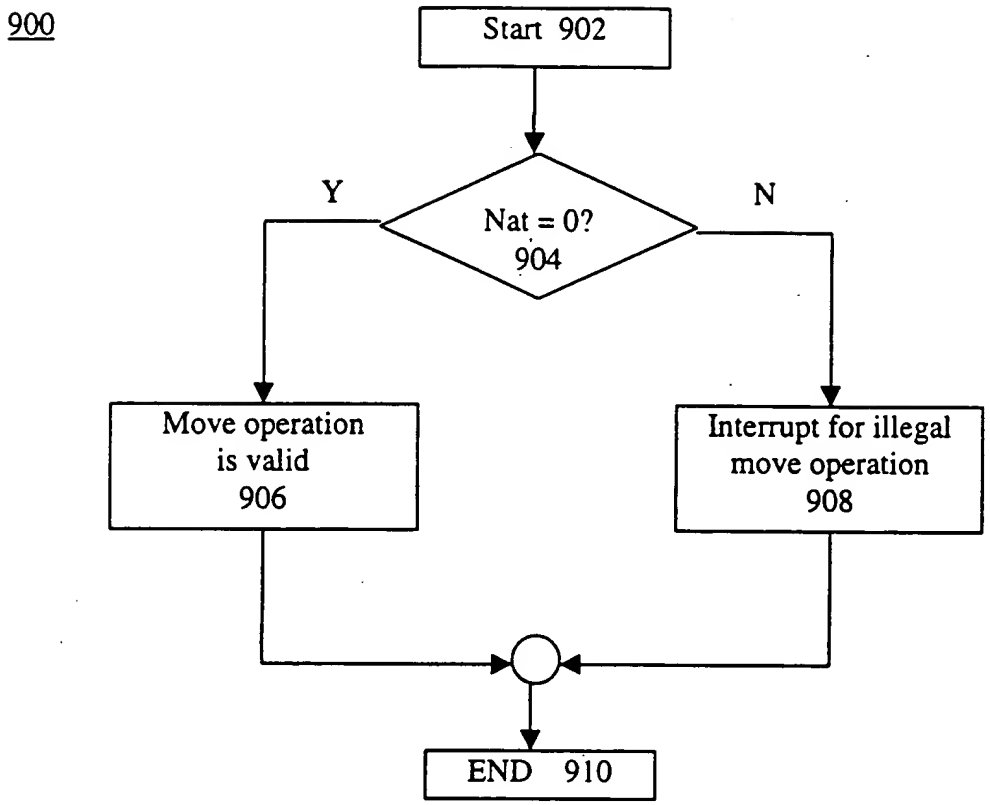
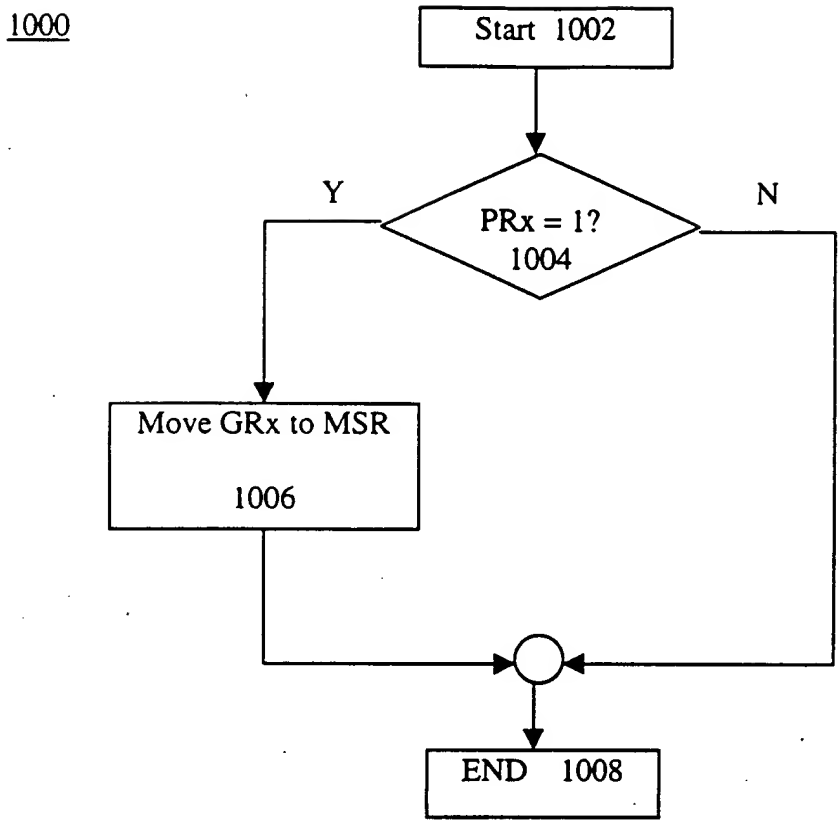

900
Start 902
Y
Nat = 0?
904
N
Move operation
is valid
906
Interrupt for illegal
move operation
908
END 910

FIG. 9



(PRx) mov MSR[index] = GRx
1010

FIG. 10

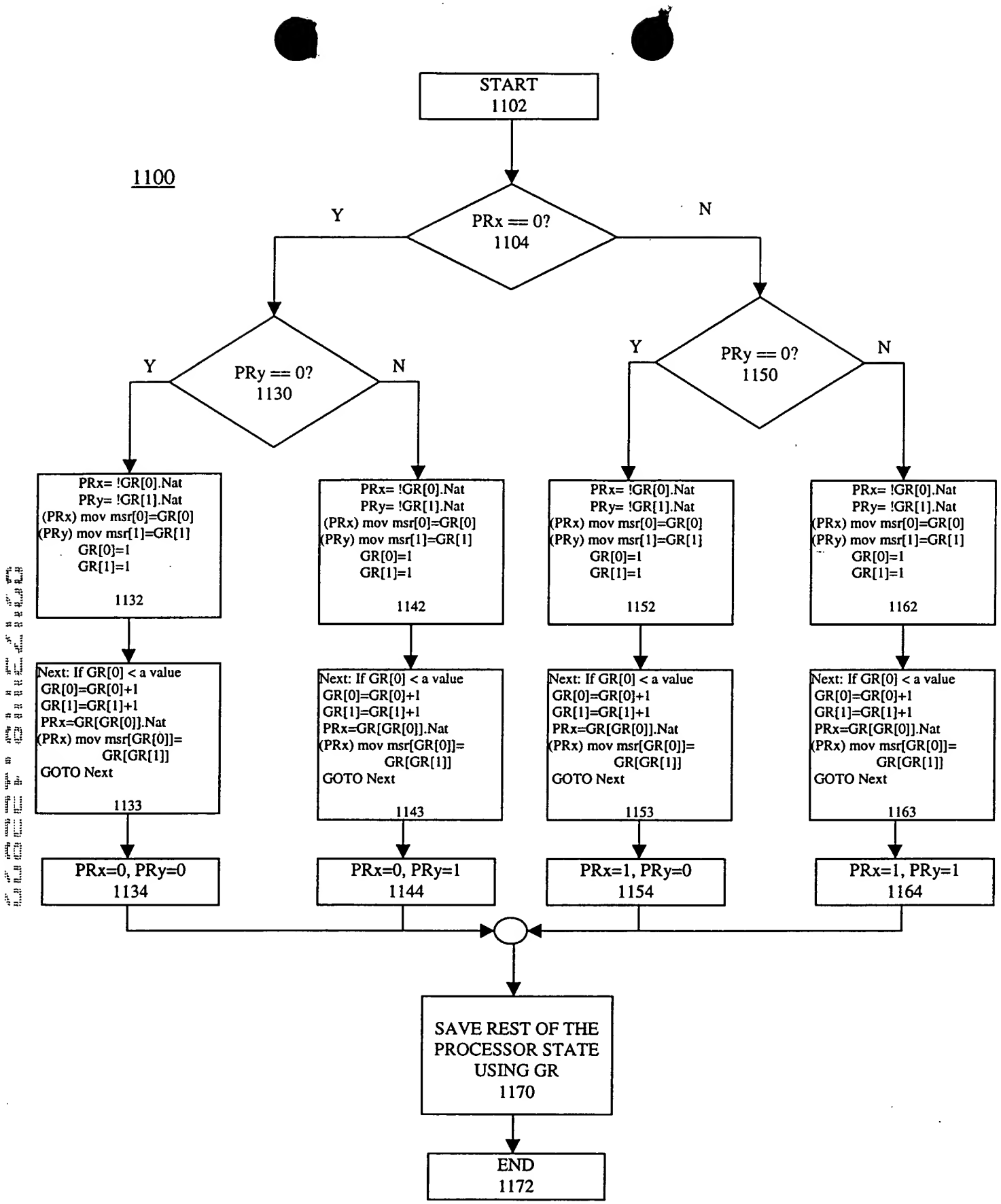

1100
START
1102
PRx == 0?
1104
Y
N
PRy == 0?
1130
PRy == 0?
1150
PRx= !GR[0].Nat
PRy= !GR[1].Nat
(PRx) mov msr[0]=GR[0]
(PRy) mov msr[1]=GR[1]
GR[0]=1
GR[1]=1
1132
1142
1152
1162
Next: If GR[0] < a value
GR[0]=GR[0]+1
GR[1]=GR[1]+1
PRx=GR[GR[0]].Nat
(PRx) mov msr[GR[0]]=
GR[GR[1]]
GOTO Next
1133
1143
1153
1163
PRx=0, PRy=0
1134
PRx=0, PRy=1
1144
PRx=1, PRy=0
1154
PRx=1, PRy=1
1164
SAVE REST OF THE
PROCESSOR STATE
USING GR
1170
END
1172

FIG. 11